И. Л. Андронов, *г. Одесса*

ОДА ФОРТУ

Впервые реализация Форта для БК-0010 появилась в 1986 г. трудами ленинградцев М. Ефимова и А. Цаплева. По тем временам для пользователей БК-0010 это был настоящий прорыв замкнутого круга под названием «Фокал». Однако первые радостные голоса быстро утихли, когда выяснилось, что Форт на БК-0010 оставляет пользователю всего лишь 3–3.5 Кб памяти, чего может хватить разве только для учебно-познавательных целей. Причина этого – ставший притчей во языцех мизерный ресурс ОЗУ БК-0010. Поэтому Форт для БК-0010 версии 1986 года не получил распространения.

Все другие причины непопулярности Форта, в том числе и его необычность, я считаю необоснованными. Утверждать обратное – значит подвергать сомнению «расторопность» пользователей БК-0010, которые одолели и не такие «необычности».

Еще чаще говорят о сложности освоения Форта, но об этом почти всегда можно услышать только от тех, кто и не пытался серьезно им овладеть. Вообще же, с учетом появившейся литературы о Форте и на основании личного опыта, смею утверждать, что Форт в освоении легче, чем Бейсик.

Что же касается «необычности», то этот термин говорит сам за себя: чего не знаешь, то и необычно. Если же разобраться по существу, то главная необычность Форта заключена в стереотипе мышления программирующего на императивных языках. По мере «вхождения» в Форт мышление быстро адаптируется.

Какими же «плюсами» отличается Форт от остальных языков?

- *Широта и глубина мышления программирующего на Форте не ограничена закостенелыми формами. «Кирпичики», с помощью которых строится программа, весьма разнообразны как по количеству, так и по диапазону сложности. Более того, лексикон этих «кирпичиков» зависит исключительно от программиста и им же обогащается, так как Форт по своей природе расширяемый язык.*
- *После создания каждого нового определения программист легко «прощается» с деталями, освобождая внимание для перехода на более высокий структурный уровень.*
- *Форт одинаково пригоден как для системного, так и для прикладного программирования.*
- *Форт для БК-0011 делает компьютер вполне «серьезной» и «самостоятельной» машиной. Не секрет, что работа на БК-0011 до сих пор шла в основном по пути не всегда эффективного использования богатого программного обеспечения от БК-0010. А вот поместив Форт на БК-0011, пользователь получает в свои руки все ресурсы машины — и «HARDWARE» и «SOFTWARE».*
- *Форт устроен так, что легко приспосабливает для своих нужд «чужое» программное обеспечение. Так, в описываемой версии (VER 2.0 JAN 92), имеется слово «BLOAD», позволяющее подгружать к Форту с магнитофона «чужие» программы в позиционно независимом коде (файлы, созданные «Меломаном», дезассемблер DISASM М. Цыпина и т.д.), а пакет арифметики с плавающей запятой напрямую использует подпрограммы из ПЗУ Бейсика.*
- *Пакет Форт-ассемблер создает интерактивную среду, более удобную, нежели при работе с MIRAGE, ОТЛ12 или, тем более, при работе в системах «MICRO». При этом Форт-ассемблер позволяет применять как локальные (в пределах одного определения), так и глобальные метки, где последними служат обычные Форт-слова (длиной до 31 символа).*
- *Форт имеет средства чтения EDASP-текстов.*
- *В Форте всегда присутствует свой редактор, по мощности сопоставимый с редактором EDASP.*
- *Если компилятор Бейсика долгое время был «закрытой зоной» для пользователей и лишь недавно «вскрыт» расторопными умельцами, то Форт сразу объявляет себя открытым для всех, даже для слабо подготовленных пользователей. Форт «насквозь просвечивается» такими мощными средст-*

вами самодокументирования, как REFOPTH, DECOMPILER, DUMP.

- *Сколько «слез пролито» программирующими на Бейсике БК-0010.01 из-за отсутствия в системе того или иного оператора (MERGE, SPRITE, MAT и др.)! А суть программирования на Форте как раз и заключается в создании (определении) новых нужных в данный момент слов-операторов, так что в некотором смысле можно говорить о неисчерпаемости Форта.*

А чем еще необычен Форт? Вот несколько примеров:

- *В Форте возможно целочисленное умножение таких величин: OCT 377 400 * (В Бейсике при этом произошло бы переполнение).*
- *Форт позволяет напечатать результат в нужном вам формате:*

1) OCT 377 400 * .
OCT 377 400 * U.
Ответы соответственно будут -400
и 177400.

2) Поместим два кода символов в одно слово:
OCT ASCII Щ 400 * ASCII Я + U. <ВК>
176761
Теперь введите такую строку:
176761 SP @ 2 TYPE DROP
ЯЩ

('DROP' здесь нужно лишь для того, чтобы не «засорять» стек данных, так как 'TYPE' не предназначено для печати со стека и поэтому данные (176761) остаются на стеке после 'TYPE').

Необычностей в Форте много, и среди них есть такие экзотические, что не отпугивают, а скорее завораживают своими чудесными и ценными свойствами. Что вы, например, скажете о возможности в Форте присваивать переменным не только величины, но и действия, т. е. переменная в некотором роде может стать оператором, и таким, каким вам заблаговолится!? Вот пример программы:

```
OKT VAR A VAR B VAR C ' NOOP C !
: ANSWER A @ B @ C @ EXECUTE ;

: SUM     ['] + C ! ANSWER ;
: DIFFER ['] - C ! ANSWER ;
: MUL     ['] * C ! ANSWER ;
: DIV     ['] / C ! ANSWER ;

:QUADR_SUM    SUM    DUP * ;
: QUADP_DIFFER DIFFER DUP * ;

10 A ! 4 B !
```

(В зависимости от вводимого вами слова переменной 'С' присваивается соответствующий оператор, который исполняется словом 'EXECUTE').

Поговорим теперь о сложности освоения Форта. Да, Форт действительно достаточно сложен. Когда я впервые прочитал, что Форт изобретен Чарльзом Муром, то термин «изобретен» представился мне несколько высокомерным. Теперь же, когда Форт стал моим «вероисповеданием», я преклоняюсь перед гением создателя. Уровень погружения в Форт безмерен. Занимаясь Фортом уже более 3 лет, я все еще ощущаю себя юнцом, постоянно встречаясь все с новыми и новыми откровениями, а количество непонятных вещей, пожалуй, даже растет.

На следующих примерах я хочу показать, что самый консервативный учитель математики, усвоивший Бейсик «на всю оставшуюся жизнь», или самый нерадивый ученик 4–5 класса, на примере четырех-пяти игр научившийся осмысленно стучать по клавиатуре, за 15–20 минут «инструктажа» или даже самостоятельно вполне могут усвоить на приведенных примерах простейший синтаксис и метод программирования на Форте (заодно попытайтесь-ка решить эти же задачки средствами Бейсика, и сравните результат).

Пример 1.

```
: TABLE  ."СТОЛ" ;
: СТОЛ   ."TABLE";
```

и т.д. — вот у вас уже и готов англо-русский и русско-английский словарь.

Форт позволяет «открытым текстом», в том числе и на русском языке, записывать (а в этом языке программирования правильно записать – значит уже решить) условия задачи, оформляя их как «определения через двоеточие». Запустив на исполнение последнее слово '??', вы получите на своем экране строку с ответом:

```
: частей_алюминия       19 ;
: частей_магния          2;
: вес_разности          34;
: разность_частей       частей_алюминия частей_магния   - ;
: вес_одной_части       вес_разности  разность_частей   / ;
: вес_алюминия          частей_алюминия вес_одной_части  * ;
: вес_магния            частей_магния   вес_одной_части  * ;
: ??   ."               ответ: вес сплава = "
                        вес_алюминия вес_магния + . ." кг" ;

??
ответ: вес сплава = 42 кг
```

Обратите внимание: мы не использовали ни одной переменной. В решении этой задачи Форт показал нам еще одну свою грань: Форт – это язык спецификаций. (В данном случае для решения задачи мы обошлись целочисленной арифметикой. Синтаксис при использовании арифметики с плавающей запятой тот же).

И в заключение хочу отметить следующее. Еще одно «препятствие», по мнению многих оппонентов (постфиксная запись формул) также является, на мой взгляд, надуманным. Во-первых Форт – стековая система, поэтому когда вы научитесь думать по-фортовски (а для этого надо немедленно начать программировать на Форте), постфиксная запись представится вам вполне естественной (кроме оправданных исключений, этот принцип прослеживается во всем словаре Форта). А во-вторых, весьма просто переопределить способ ввода на любой (но нужно ли это?). Вот пример для инфиксной записи:

```
:INFIX BL WORD NUMBER DROP ;
```

Теперь можно переопределить любые операторы:

```
: + INFIX + ;
: - INFIX - ;
............
: AND INFIX AND ;
```

и т.д.

И теперь можно вводить строки формул «обычным» способом (без скобок!).

Вообще говоря, возможности Форта, как тема для разговора, не имеют видимых границ, и вы наверняка откроете для себя много нового в чудесном мире Форта.